

The RS-tree: An Efficient Data Structure for Distance Browsing Queries *

Dong-Joo Park[†] Shin Heu[‡] Hyoung-Joo Kim[†]

[†]School of Computer Science and Engineering [‡]Department of Computer Science and Engineering
Seoul National University, Korea Hanyang University, Korea

{djpark, hjk}@oopsla.snu.ac.kr

shinheu@cse.hanyang.ac.kr

Keywords: multimedia database, multi-dimensional index, k -nearest neighbor query, signature file

1 Introduction

Recently, multimedia applications such as geographic information systems(e.g., Paradise[9] and Dedale[5]) and image retrieval systems(e.g., QBIC[4] and Chabot[7]) require the efficient processing of k -nearest neighbor queries over a collection of d -dimensional spatial objects. *Distance browsing queries* like these frequently involve non-spatial predicates as well as spatial proximity. For example, the query “choose k albums by the Beatles whose cover colors best match red” (in short, $\text{artist}='Beatles' \wedge \text{color}='red'$), carries the spatial predicate $\text{color}='red'$ and the non-spatial predicate $\text{artist}='Beatles'$.

Among many available techniques for such queries, Hjaltason and Samet’s incremental nearest neighbor algorithm is known as the most efficient algorithm[6]. The reason is that when the upper operator(for example, for $\text{artist}='Beatles'$) needs another TID(i.e., a tuple ID)¹ of tuple containing the next neighbor, their algorithm can provide it without restarting the query from scratch[6]. However, their algorithm may generate a large number of worthless tuples(or TIDs) that will turn out not to satisfy the remaining non-spatial predicates, leading it to inefficiency. In the study, we propose an *RS-tree* and a new algorithm based on it, which is *complementary* to Hjaltason and Samet’s algorithm, in the sense that it can partially prune worthless tuples as early as possible.

In this paper, we assume that for simplicity, the query involves one equal match predicate. Our approach also is under the assumption that the selectivity of the non-spatial predicate is so *high*² that the query optimizer decides to use the spatial index(as Hjaltason and Samet’s approach is implicitly[6]).

2 R-tree Incremental Nearest Neighbor Algorithm

Hjaltason and Samet[6] presented an incremental nearest neighbor algorithm³ based on hierarchical multidimensional indexes, especially the R-tree(from now on, *RtreeINN*). The *RtreeINN* algorithm first initializes the priority queue. Then the algorithm traverses the root node in the R-tree, and for each child node in the

*This work was supported in part by University Research Program supported by Ministry of Information & Communication in South Korea, entitled “A Study on Extending Spatial Database and Its Application to Spatial Data Warehousing” and in part by the Brain Korea 21 Project.

¹It is assumed that each leaf node in the hierarchical index stores a pair of a spatial object and a TID.

²It means that relatively a large number of tuples satisfy the predicate.

³The algorithm assumes the case where the R-tree stores d -dimensional *points*. A general algorithm for any spatial object(e.g., polygon, polyline, point, and so on) was described in [6].